



```
namespace Arbol
{
    public partial class FormArbol : Form
    {
        Arbol A;
        public FormArbol()
        {
            A = new Arbol();
            InitializeComponent();
        }
        private void btnInsertar_Click(object sender, EventArgs e)
        {
            int dato = Convert.ToInt32(txtNum.Text);
            //int dato = Int.Parse(txtNum.Text);
        }
    }
}
```

```
        MessageBox.Show(A.insertar(dato));
        txtNum.Clear();
        txtNum.Focus();
    }

    public void Mostrar_preOrden()
    {
        Nodo raiz = A.devolverRaiz();
        Mostrar_preOrden(raiz);
    }
    private Nodo Mostrar_preOrden(Nodo raiz) {
        if (raiz != null)
        {
            LstPreorden.Items.Add(raiz.Dato);
            Mostrar_preOrden(raiz.RamaIzq);
            Mostrar_preOrden(raiz.Ramader);
        }
        return raiz;
    }
    private Nodo Mostrar_InOrden(Nodo raiz){
        if(raiz != null)
        {
            Mostrar_InOrden(raiz.RamaIzq);
            LstInOrden.Items.Add(raiz.Dato);
            Mostrar_InOrden(raiz.Ramader);
        }
        return raiz;
    }
    public void Mostrar_PosOrden()
    {
        Nodo raiz = A.devolverRaiz();
        Mostrar_PosOrden(raiz);
    }
    private Nodo Mostrar_PosOrden(Nodo raiz)
    {
        if(raiz != null)
        {
            Mostrar_PosOrden(raiz.RamaIzq);
            Mostrar_PosOrden(raiz.Ramader);
            LstPostOrden.Items.Add(raiz.Dato);
        }
        return raiz;
    }
    private void btoInOrd_Click(object sender, EventArgs e)
    {
        Nodo raiz = A.devolverRaiz();
        Mostrar_preOrden(raiz);
        Mostrar_InOrden(raiz);
        Mostrar_PosOrden(raiz);
    }

    private void btoBuscar_Click(object sender, EventArgs e)
    {
        int bus = Convert.ToInt32(txtNum.Text);
        Nodo Aux = A.BuscarNodo(bus);
        if(Aux != null)
        {
            MessageBox.Show("Nodo: " + bus + " encontrado");
        }
        else
        {
            MessageBox.Show("El nodo que buscas no existe");
        }
    }
}
```

```
    }  
}  
  
private void btoEliminar_Click(object sender, EventArgs e)  
{  
    int elim = Convert.ToInt32(txtNum.Text);  
    bool eliminado = A.Eliminar(elim);  
    if (eliminado)  
    {  
        MessageBox.Show("El Nodo: " + elim + " fue eliminado");  
    }  
    else  
    {  
        MessageBox.Show("El nodo que buscas no existe");  
    }  
}  
  
private void btoLimpiar_Click(object sender, EventArgs e)  
{  
    LstInOrden.Items.Clear();  
    LstPostOrden.Items.Clear();  
    LstPreOrden.Items.Clear();  
}  
  
private void bto_IOr_Click(object sender, EventArgs e)  
{  
    Nodo raiz = A.devolverRaiz();  
    if (raiz != null)  
    {  
        Mostrar_InOrden(raiz);  
    }  
    else  
    {  
        MessageBox.Show("El Arbol esta Vacio");  
    }  
}  
  
private void bto_RecPre_Click(object sender, EventArgs e)  
{  
    Nodo raiz = A.devolverRaiz();  
    if(raiz != null)  
    {  
        Mostrar_preOrden(raiz);  
    }  
    else  
    {  
        MessageBox.Show("El Arbol esta Vacio");  
    }  
}  
  
private void btoPost_Click(object sender, EventArgs e)  
{  
    Nodo raiz = A.devolverRaiz();  
    if (raiz != null)  
    {  
        Mostrar_PosOrden(raiz);  
    }  
    else  
    {  
        MessageBox.Show("El Arbol esta Vacio");  
    }  
}
```

```
}
```

```
namespace Arbol
{
    class Arbol
    {
        Nodo raiz;
        Nodo Nuevo;
        public Arbol()
        {
            raiz = null;
            Nuevo = null;
        }
        public string insertar(int dato)
        {
            Nuevo = new Nodo(dato);
            if(raiz == null)
            {
                raiz = Nuevo;
                return "El Nodo fue insertado";
            }
            else
            {
                Nodo Aux = raiz;
                while(Aux != null)
                {
                    if (Nuevo.Dato > Aux.Dato)
                    {
                        if (Aux.Ramader == null)
                        {
                            Aux.Ramader = Nuevo;
                            return "El Nodo fue insertado Lado Derecho";
                        }
                        Aux = Aux.Ramader;
                    }
                    else
                    {
                        if (Aux.RamaIzq == null)
                        {
                            Aux.RamaIzq = Nuevo;
                            return "El Nodo fue insertado Lado Izquierdo";
                        }
                        Aux = Aux.RamaIzq;
                    }
                }
                return "";
            }
        }
        public Nodo BuscarNodo(int dato)
        {
            if (raiz != null)
            {
                Nodo Aux = raiz;
                while (Aux.Dato != dato)
                {
                    if (dato < Aux.Dato)
                    {
                        Aux = Aux.RamaIzq;
                    }
                    else
                    {

```

```
        Aux = Aux.Ramader;
    }
    if (Aux == null)
    {
        return null;
    }
    return Aux;
}
else
    return null;
}
public bool Buscar(int dato)
{
    Nodo Aux = Buscar(dato, raiz);
    if(Aux != null)
    {
        return true;
    }
    else
    {
        return false;
    }
}
//Modo Recursivo
private Nodo Buscar(int dato, Nodo Aux)
{
    if (Aux != null)
    {
        if (Aux.Dato != dato)
        {
            if (dato < Aux.Dato)
            {
                Aux = Aux.RamaIzq;
                Buscar(dato, Aux);
            }
            else
            {
                Aux = Aux.Ramader;
                Buscar(dato, Aux);
            }
        }
        return Aux;
    }
    else
    {
        return null;
    }
}
public Nodo BorrarMenor()
{
    Nodo Aux = null;
    if (raiz == null)
    {
        return null;
    }
    else
    {
        if (raiz != null)
        {
            if (raiz.RamaIzq == null)
            {

```

```
        raiz = raiz.Ramader;
    }
    else
    {
        Nodo Atras = raiz;
        Aux = raiz.RamaIzq;

        while (Aux.RamaIzq != null)
        {
            Atras = Aux;
            Aux = Aux.RamaIzq;
        }
        Atras.RamaIzq = Aux.Ramader;
    }
}
return Aux;
}

public Nodo EliminarMayor()
{
    Nodo Aux = null;
    if (raiz == null)
    {
        return null;
    }
    else
    {
        if (raiz != null)
        {
            if (raiz.Ramader == null)
            {
                raiz = raiz.RamaIzq;
            }
            else
            {
                Nodo Atras = raiz;
                Aux = raiz.Ramader;

                while (Aux.Ramader != null)
                {
                    Atras = Aux;
                    Aux = Aux.Ramader;
                }
                Atras.Ramader = Aux.RamaIzq;
            }
        }
    }
    return Aux;
}

public bool Eliminar(int dato)
{
    if(raiz == null)
    {
        return false;
    }
    else
    {
        Nodo Aux = raiz;
        Nodo Padre = raiz;
        bool HijoIzq = true;
```

```
while(Aux.Dato != dato)
{
    Padre = Aux;
    if(dato < Aux.Dato)//Si el Datos es menor que la raiz se va a
la izquierda
    {
        HijoIzq = true;
        Aux = Aux.RamaIzq;
    }
    else
    {
        HijoIzq = false;
        Aux = Aux.Ramader;
    }
}
if(Aux.RamaIzq == null && Aux.Ramader == null)//Si el Nodo es
Hoja o unico
{
    if(Aux == raiz)//Si el Nodo a eliminar es la raiz
    {
        raiz = null;
    }
    else if (HijoIzq)//Si el nodo a eliminar es hoja y es menor a
la raiz
    {
        Padre.RamaIzq = null;
    }
    else//Si es hoja y mayor a la raiz
    {
        Padre.Ramader = null;
    }
}
else if(Aux.Ramader == null)//Si el dato a eliminar no es hoja y
tiene rama izquierda
{
    if(Aux == raiz)//Si el nodo a eliminar es la raiz
    {
        raiz = Aux.RamaIzq;
    }
    else if (HijoIzq)
    {
        Padre.RamaIzq = Aux.RamaIzq;
    }
    else
    {
        Padre.Ramader = Aux.RamaIzq;
    }
}
else if(Aux.RamaIzq == null)//Si el dato a eliminar no es hoja y
tiene rama derecha
{
    if(Aux == raiz)
    {
        raiz = Aux.Ramader;
    }
    else if (HijoIzq)
    {
        Padre.RamaIzq = Aux.Ramader;
    }
    else
    {

```

```
        Padre.Ramader = Aux.RamaIzq;
    }
}
else//Si el nodo a eliminar no es hoja , tampoco tiene unica rama
{
    Nodo Reemplazo = ObtenerReemplazo(Aux);
    if(Aux == raiz)
    {
        raiz = Reemplazo;
    }
    else if (HijoIzq)
    {
        Padre.RamaIzq = Reemplazo;
    }
    else
    {
        Padre.Ramader = Reemplazo;
    }
    Reemplazo.RamaIzq = Aux.RamaIzq;
}
}
return true;
}

public Nodo ObtenerReemplazo(Nodo r)
{
    Nodo ReemplazoPadre = r;
    Nodo Reemplazo = r;
    Nodo Aux = r.Ramader;

    while(Aux != null)
    {
        ReemplazoPadre = Reemplazo;
        Reemplazo = Aux;
        Aux = Aux.RamaIzq;
    }

    if(Reemplazo != r.Ramader)
    {
        ReemplazoPadre.RamaIzq = Reemplazo.Ramader;
        Reemplazo.Ramader = r.Ramader;
    }
    return Reemplazo;
}

public Nodo devolverRaiz()
{
    return raiz;
}
}

class Nodo
{
    int dato;
    Nodo Arbol_pun_izq;
    Nodo Arbol_pun_der;
    public Nodo(int d)
    {
        Dato = d;
        Ramader = null;
        RamaIzq = null;
    }
}
```



```
public Nodo()  
{  
    Dato = 0;  
    Ramader = null;  
    RamaIzq = null;  
}  
public int Dato { get => dato; set => dato = value; }  
public Nodo RamaIzq { get => Arbol_pun_izq; set => Arbol_pun_izq = value;  
}  
public Nodo Ramader { get => Arbol_pun_der; set => Arbol_pun_der = value;  
}  
}
```